Combining fast_align with Hierarchical Sub-sentential Alignment for Better Word Alignments

Hao Wang
Graduate School of Information, Production and Systems, Waseda University
oko_ips@ruri.waseda.jp

Yves Lepage
Graduate School of Information, Production and Systems, Waseda University
yves.lepage@waseda.jp

Abstract

fast_align is a simple and fast word alignment tool which is widely used in state-of-the-art machine translation systems. It yields comparable results in the end-to-end translation experiments of various language pairs. However, fast_align does not perform as well as GIZA++ when applied to language pairs with distinct word orders, like English and Japanese. In this paper, given the lexical translation table output by fast_align, we propose to realign words using the hierarchical sub-sentential alignment approach. Experimental results show that simple additional processing improves the performance of word alignment, which is measured by counting alignment matches in comparison with fast_align. We also report the result of final machine translation in both English-Japanese and Japanese-English. We show our best system provided significant improvements over the baseline as measured by BLEU and RIBES.

1 Introduction

Since state-of-the-art machine translation systems start with word aligned data, the processing of word alignment plays a fundamental role in machine translation. A reliable and accurate word aligner is considered as an essential component in the various implementations of machine translation, e.g., word-based model (Brown et al., 1990), phrase-based model (Koehn et al., 2003), hierarchical phrase-based model (Chiang, 2005) and tree-to-tree model (Gildea, 2003; Zhang et al., 2007). In general, word alignment is prerequisite for extracting rules or sub-translations (word pairs, phrase pairs or partial tree templates) for translation.

The most widely used word aligner is GIZA++ (Och and Ney, 2000), which is based on *generative* models, like IBM models (Brown et al., 1993) and HMM-based model (Vogel et al., 1996), in which parameters are estimated using the Expectation-Maximization (EM) algorithm. This generative approach allows GIZA++ to automatically extract bilingual lexicon from parallel corpus without any annotated data. Besides, a variation of IBM model 2 was implemented as fast_align[1] (Dyer et al., 2013), which allows an effective alignment of words. There is no doubt that fast_align is almost the fastest word aligner, while keeping the quality of alignment, compared to the baseline using GIZA++[2] (Och and Ney, 2003), or MGIZA++[3] (Gao and Vogel, 2008).

However, Ding et al. (2015) demonstrated that fast_align does not outperform the baseline GIZA++, especially for the distantly related language pairs, like English-Japanese or Chinese-English. The reason may be explained by the fact that, given a source word, fast_align tends to limit the probable target translation and its alignment nearest as possible to the diagonal in the alignment matrix according to the overall word orders, which is the drawback of IBM-model 2 (Brown et al., 1993) and its variations, in terms of being insensitive to word orders. The word alignments output by fast_align

[1] https://github.com/clab/fast_align
[2] http://www.statmt.org/moses/giza/GIZA++.html
[3] http://www.cs.cmu.edu/~qing/giza/

Proceedings of the Sixth Workshop on Hybrid Approaches to Translation, pages 1–7, Osaka, Japan, December 11, 2016.

are often more compact represented in alignment matrices. For the case of distinct language pairs, this strategy damages the quality of the final alignment result.

Since IBM Model is restriction of one-to-many (1-m) alignments, some multi-word units cannot be correctly aligned. It is necessary to train models in both directions, and merge the outcome of mono-directional alignments using some symmetrization methods, for example, *grow-diag-final-and* (Och and Ney, 2003). Though this method can overcome the mentioned deficiency to some degree, the strong assumption of 1-m alignment forces the aligner to generate 1-best alignments, which is prone to learn noisy rules due to alignment or segmentation mistakes. Another problem exists is that the production of 1-m alignment losses the structural information of the whole sentence while phrase-based (or other kinds of statistical machine translation systems) relies on the continuous translation fragments. It has been proved that by applying structural models such as Inversion Transduction Grammars (ITG) (Wu, 1997) will achieve some gain. ITG has been widely applied to word alignment, bilingual parsing, etc., due to its simplicity and effectiveness of modeling bilingual correspondence. However, inducing ITGs from parallel data would be time-consuming.

In this paper, in order to integrate ITG with IBM model, we propose to apply the hierarchical sub-sentential alignment (HSSA) (Lardilleux et al., 2012) approach to realign word alignments. HSSA is an online word alignment approach, which was first introduced as complementary to `Anymalign`[4]. When fed with the lexical weights output by `Anymalign`, it yields comparable results with baseline MGIZA++. In fact, an important advantage of this approach is that it can be combined with any other existing approach by reusing the lexical weights output by this other approach. We make use of the structure named soft alignment matrix (Liu et al., 2009) to represent the alignment distribution for a given sentence pair, which cells are weighted by the lexical weights output by `fast_align`. With the recursive binary segmentation processing in HSSA, we realign the sentence pairs top-down. We also present a simple but effective method to deal with error alignment points produced by this hybrid method, i.e., conflicting cells in soft alignment matrices.

In Section 2 and Section 3, the notion of soft alignment matrix and HSSA will be introduced. The hybrid combination architecture of our proposed method will be illustrated in Section 4. Experimental results and the analysis will be given in the following Section 4. Finally, Section 6 draws the conclusion and future work.

2 Soft Alignment Matrices

A sentence pair matrix can be interpreted as a contingency matrix for the source sentence f (length J) relatively to the target sentence e (length I). Formally, given a source sentence $f = f_1^J = f_1, \ldots, f_j, \ldots, f_J$ and a target sentence $e = e_1^I = e_1, \ldots, e_i, \ldots, e_I$, we define a soft link $l = (j, i)$ to exist if f_j and e_i are probable translation. Then, given the word positions (j, i) in a $J \times I$ soft alignment matrix, $\mathcal{M}(J, I)$, a score w for each cell $\mathcal{M}(i, j)$ is definded as:

$$w(j,i) = \begin{cases} \alpha & \text{if } l = \varepsilon \\ \sqrt{p(f_j|e_i) \times p(e_i|f_j)} & \text{otherwise} \end{cases} \quad (1)$$

where w measures the strength of the translation link[5] between any source and target pair of words (f_j, e_i), in our case, the score $w(f_j, e_i)$ is defined as the geometric mean of the bidirectional lexical translation probabilities. The symmetric alignment between word f_j and e_i is visualized as a greyed cell $\mathcal{M}(i, j)$ in this matrix (see Figure 1). For example, the word pair ("japanese", "日本") is definitely aligned, but ("ink", "日本") is definitely unaligned.

In fact, the resulting soft alignment matrix makes it possible to refine the final output of alignments and reduce alignment errors. Since sub-sentential alignment interests us more than single word-to-word alignment, we define a score for phrasal case. Differing to the defination of phrase translation probability

[4] `https://anymalign.limsi.fr/`

[5] To avoid problems linked with data sparsity, Laplace smoothing was used here to handle the unseen alignments, with assigned a very small smoothing parameter $\alpha = 10^{-7}$.

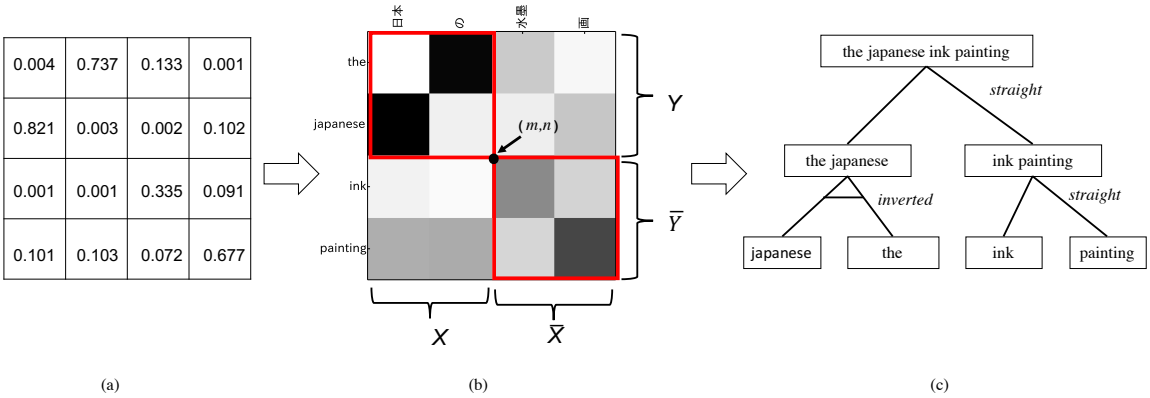

Figure 1: (a) A soft alignment matrix; (b) the grey-scale graph of soft alignment matrix; (c) corresponding ITG parsing tree. In Figure (a), cells are greyed from 0.0 (white) to 1.0 (black) on a logarithmic scale.

and lexical weighting (Koehn et al., 2003), the score of a block (X, Y) is defined as the summation w of the association scores between each source and target word pair inside this block as (Matusov et al., 2004; Lardilleux et al., 2012):

$$W(X, Y) = \sum_{f \in X} \sum_{e \in Y} w(f, e) \tag{2}$$

We employ the structure of summed area table for quick computation of the score $W(X, Y)$ in a $O(1)$ time complexity. Hereby, normalization of the probability distribution is not necessary. It should be emphasized that our soft matrix is estimated differing with the weighted matrix in (Liu et al., 2009).

3 Hierarchical Sub-sentential Alignment Approach

Given the soft alignment matrix, the HSSA approach takes all cells in the soft alignment matrix into consideration and seeks the precise criterion for a good partition in a similar way as image segmentation. HSSA makes use of an unsupervised clustering algorithm called *normalized cuts* (Shi and Malik, 2000), i.e., spectral clustering, or *Ncut* for short, to recursive segment the matrix into two parts. This procedure can be thought as being similar as the two rules in ITG: S (*straight*) and I (*inverted*). The ITG approach builds a synchronous parse tree for both source and target sentences, assuming that the trees have the same underlying structure (ITG tree) but that the ordering of constituents may differ in the two languages. In ITG, final derivations of sentence pairs correspond to alignments. A single non-terminal spanning a bitext cell with a source and target span corresponds to the final 1-to-many or many-to-1 HSSA alignment. In other words, HSSA performs the same kind of procedure as synchronous parsing under ITG. In ITG, there are three simple generation rules:

$$S : \gamma \rightarrow [X_1 X_2] \quad | \quad I : \gamma \rightarrow < X_1 X_2 > \quad | \quad T : \gamma \rightarrow w = (f, e) \tag{3}$$

During the segmenting, HSSA is supervised by the ITG constraint to decide the search scope of next level on the diagonal or anti-diagonal corresponding to the case of *straight* and *inverted*. HSSA terminates at the prerequisite condition when all words in source and target sentences are aligned and for each is a 1-1 alignment at least (corresponding to rule T). 1-1 means that one source word only has one aligned target word with strong confidence in both directions.

Consider a source phrase in Figure 1, $X\overline{X}$ split at index m corresponding to a target phrase $Y\overline{Y}$ split at index n. $Ncut$ is defined as (Zha et al., 2001):

$$Ncut(m, n, XY, \overline{XY}) = \frac{cut(X,Y)}{cut(X,Y)+2\times W(X,Y)} + \frac{cut(\overline{X},\overline{Y})}{cut(\overline{X},\overline{Y})+2\times W(\overline{X},\overline{Y})}$$

$$cut(X, Y) = W(X, \overline{Y}) + W(\overline{X}, Y) \tag{4}$$

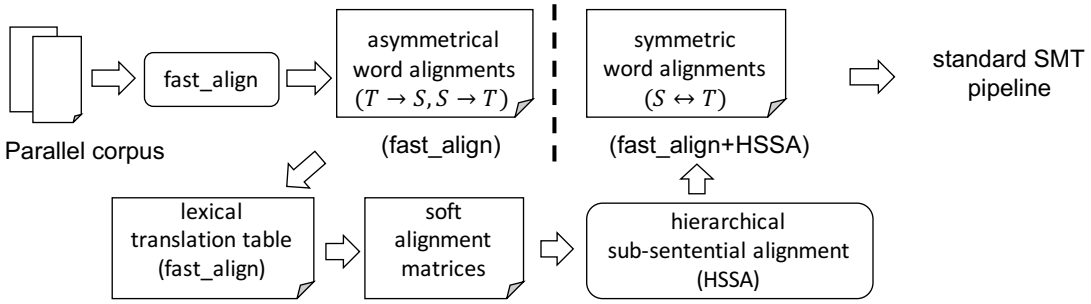

Figure 2: An example of our proposed hybrid combination architecture.

Each possible splitting point (m, n) in the matrix divides the parent matrix into 4 sub-matrices (XY, $X\overline{Y}$, $\overline{X}Y$, $\overline{XY}$). Either the two sub-matrices on the diagonal (XY, $\overline{XY}$) or the two sub-matrices on the anti-diagonal ($X\overline{Y}$, $\overline{X}Y$) will be chosen to limit the search scope on the next level. Hence, recursive segmentation eventually consists in determining the indices (m, n) which minimizes $Ncut(m, n,XY,\overline{XY})$ or $Ncut(m, n,\overline{X}Y,X\overline{Y})$ over all possible indices.

After computing the score of *Ncut*, HSSA decides for the next search scope (the upper left and lower right blocks in Figure 1) by finding the position where $Ncut(m, n,XY,\overline{XY})$ or $Ncut(m, n,\overline{X}Y,X\overline{Y})$ is the minimum value among all possible bipartite segmentation positions. In this example, $Ncut(m, n,XY,\overline{XY})$ is less than $Ncut(m, n,\overline{X}Y,X\overline{Y})$, equals to *straight* rule.

Since the time complexity of top-down HSSA algorithm is cubic ($O(I \times J \times \min(I, J))$, the worst case) in the length of the input sentence pair, it is faster than the original ITG approach $O(n^6)$ employing the CYK algorithm and achieves the same performance compared to (Zhang et al., 2008) which has a best time complexity of $O(n^3)$ with synchronous parsing.

4 Hybrid Combination Architecture

It is thus possible to use various word alignment tools, while `fast_align` provides the most effective pipeline with an acceptable time cost. Given the output alignments of `fast_align`, it is quite straightforward to estimate a maximum likelihood lexical translation table. We record both the direct $p(f|e)$ as well as the inverse $p(e|f)$ word translation probabilities in the translation table. This step is easy and fast finished with the `Moses`[6] training pipeline.

The purpose that drives us to do this work is the idea of combining two different models into one. One (ITG) models distinct language pair well, while the other one (IBM models) models similar language pair well. Previous work (Haghighi et al., 2009) proved that importing ITG limitations improves word alignments for Chinese-English alignment. An example illustrating our proposed hybridization is shown in Figure 2. In the context of system combination, we extend the pipeline of standard phrase-based statistical machine translation. In the middle, a soft alignment matrix (as the one in Figure 1) is generated for each sentence pair by feeding it with scores from the lexical translation table. On such soft alignment matrices, we apply the HSSA approach to obtain a final word-to-word alignment. Thanks to the simplicity of the HSSA approach, this can be done at no time cost (less than 1 minute in a real experiment on 320K sentence pairs). We employ the implementation *cutnalign*[7] for HSSA step.

Nevertheless, because HSSA outputs both 1-to-many and many-to-1 alignments, a drawback is, sometimes it returns some "noisy" alignments (referring to the alignment that appears weak in the soft alignment matrix). To solve this problem, instead of outputting all 1-to-1 matches contained in 1-to-many or many-to-1 blocks, it is better to prune low confidence matches while tweaking the alignments with heuristic search techniques, like the *grow* step in the *grow-diag-final-and* heuristic (Koehn et al., 2005). We consider that HSSA provides an alternative to *grow-diag-final-and* for alignments symmetrization in

[6] `http://www.statmt.org/moses/`
[7] `https://github.com/wang-h/min-cutnalign`

						en-ja		ja-en	
	#	MatchRef	Prec	Rec	AER	BLEU	RIBES	BLEU	RIBES
Ref	33,377								
GIZA++	31,342	18,641	59.48	55.85	42.39	**21.59**	68.10	**18.78**	65.87
fast_align	25,368	14,076	55.49	42.17	52.08	20.79‡	68.13	18.23†	65.25
+ HSSA 1-n/n-1	43,061	14,990	34.81	44.91	60.78	**21.23**	68.01	18.14†	64.91
+ prune	27,982	13,542	48.40	40.57	55.86	**21.83**	68.42	**18.38**	65.53
+ grow	30,714	13,968	45.48	41.85	56.41	**21.53**	68.14	**18.53**	65.57

Table 1: Word alignment scores on English-Japanese and translation scores (BLEU and RIBES) in both directions (English-Japanese and Japanese-English). *prune* is the case when filtering all alignments in 1-n/n-1 blocks using a threshold $\gamma > 0.001$. Boldface indicates no significantly different with GIZA++ baseline (†: $p < 0.05$, ‡: $p < 0.01$) .

replacement of the intersection alignments of fast_align. Following this idea, we produce alignments with different strategy profiles.

5 Experiments

English-Japanese alignment and translation is a much harder task for fast_align than French-English alignment (Dyer et al., 2013). In our experiments, standard phrase-based statistical machine translation systems were built by using the Moses toolkit (Koehn et al., 2007), Minimum Error Rate Training (Och, 2003), and the KenLM language model (Heafield, 2011). The default training pipeline for phrase-based SMT is adopted with default *distortion-limit* 6. Two baseline systems, one built with GIZA++ and another built with fast_align, are prepared for result comparison. For the evaluation of machine translation quality, some standard automatic evaluation metrics have been used, like BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010) in all experiments. Since BLEU is insensitive to long-distance displacements of large sequences of words, we also use RIBES which was designed to take distinct word orders into consideration. In order to ensure a consistent, repeatable and reproducible experiment, we use the original training, tuning and test sets provided in KFTT corpus[8].

We first report the performance of various alignment profiles in terms of precision, recall and alignment error rate (AER) (Och and Ney, 2003) on the basis of human annotated alignment data provided with the KFTT corpus in Table 1. The first and second lines show the alignment difference using GIZA++ and fast_align. The original HSSA, which allows 1-to-many or many-to-1 alignments, outperforms the fast_align baseline from the point view of matching alignments and recall against the reference. The total number of alignments is much higher than with fast_align which victim of the "noisy alignments" problem mentioned in Section 4. AER and precision are behind fast_align, even more than GIZA++ baseline. However, (Fraser and Marcu, 2007; Ganchev et al., 2008) question the link between this word alignment quality metrics and translation results, like whether improvements in alignment quality metrics lead to improvements in phrase-based machine translation performance.

A lower AER does not imply a better translation accuracy. We show it in the following discussion. When sampling the alignment results, we found that the output of the proposed hybrid approach usually generates better alignments than the baseline.

Experimental results in both direction for English-Japanese and Japanese-English are shown in the right part of Table 1. Specially for Japanese, we skip the particles like {*ga, wo, ha*} and remove them from the data before implementing word alignments. Translation in both direction is improved significantly over the fast_align baseline[9] in BLEU and RIBES. It is not surprising that the pruning processing performs worse on Japanese-English not as well as English-Japanese, because a single English word may be aligned with several Japanese words. Perhaps deleting low confidence alignments in the many-to-1 case impacts consistency in phrases during phrase extraction. This is why *grow* slightly

[8] http://www.phontron.com/kftt/index.html

[9] On GIZA++ experiment, HSSA decreases in the final translation score somehow.

improved the final translation result.

6 Conclusion

This work presented a hybrid application of the hierarchical sub-sentential alignment approach with `fast_align`. It can be seen as an attempt to import the ITG framework into the IBM models. We showed that through the simple additional processing, our proposed approach yields better results than baselines. We also demonstrate that given reliable values, the heuristic alignment method based on word association (Moore, 2005) could yield competitive results with more complex parameter estimation approaches.

Acknowledgments

This work is supported in part by China Scholarship Council (CSC) under the CSC Grant No.201406890026. We also thank the anonymous reviewers for their insightful comments.

References

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 2015. Improving fast align by reordering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal*. Citeseer.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3).

Kuzman Ganchev, Joao V Graça, and Ben Taskar. 2008. Better alignments= better translations? *ACL-08: HLT*, page 986.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 80–87. Association for Computational Linguistics.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 923–931. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *IWSLT*, pages 68–75.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Adrien Lardilleux, François Yvon, and Yves Lepage. 2012. Hierarchical sub-sentential alignment with anymalign. In *16th annual conference of the European Association for Machine Translation (EAMT 2012)*, pages 279–286.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 1017–1026. Association for Computational Linguistics.

Evgeny Matusov, Richard Zens, and Hermann Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 219. Association for Computational Linguistics.

Robert C Moore. 2005. Association-based bilingual word alignment. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 1–8. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. Giza++: Training of statistical translation models.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. 2001. Bipartite graph partitioning and data clustering. pages 25–32.

Min Zhang, Hongfei Jiang, AiTi Aw, Jun Sun, Sheng Li, and Chew Lim Tan. 2007. A tree-to-tree alignment-based model for statistical machine translation. *MT-Summit-07*, pages 535–542.

Hao Zhang, Chris Quirk, Robert C Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *ACL*, pages 97–105.